

A Generalization of the 2D-DSPM for Solving Linear System of Equations

Davod Khojasteh Salkuyeh

Department of Mathematics, University of Mohaghegh Ardabili,

P. O. Box. 56199-11367, Ardabil, Iran

E-mail: khojaste@uma.ac.ir

Abstract

In [N. Ujević, New iterative method for solving linear systems, Appl. Math. Comput. 179 (2006) 725730], a new iterative method for solving linear system of equations was presented which can be considered as a modification of the Gauss-Seidel method. Then in [Y.-F. Jing and T.-Z. Huang, On a new iterative method for solving linear systems and comparison results, J. Comput. Appl. Math., In press] a different approach, say 2D-DSPM, and more effective one was introduced. In this paper, we improve this method and give a generalization of it. Convergence properties of this kind of generalization are also discussed. We finally give some numerical experiments to show the efficiency of the method and compare with 2D-DSPM.

AMS Subject Classification : 65F10.

Keywords: linear system, projection method, Gauss-Seidel method, Petrov-Galerkin condition, convergence.

1. Introduction

Consider the linear system of equations

$$Ax = b, \tag{1}$$

where $A \in \mathbb{R}^{n \times n}$ is a symmetric positive definite (SPD) matrix and $x, b \in \mathbb{R}^n$. The Gauss-Seidel method is an stationary iterative method for solving linear system of equation and is convergent for SPD matrices. This method is frequently used in science and engineering, both for solving linear system of equations and preconditioning [2, 6]. It can be easily seen that the Gauss-Seidel method is an special case of

a projection method [1, 6]. Let \mathcal{K} and \mathcal{L} be two m -dimensional subspaces of \mathbb{R}^n . Let also x_0 be an initial guess of the solution of (1). A projection method onto \mathcal{K} and orthogonal to \mathcal{L} is a process which finds an approximate solution $x \in \mathbb{R}^n$ to (1) by imposing the conditions that $x \in x_0 + \mathcal{K}$ and the new residual vector be orthogonal to \mathcal{L} (Petrov-Galerkin condition), i.e.

$$\text{Find } x \in x_0 + \mathcal{K}, \quad \text{such that} \quad b - Ax \perp \mathcal{L}. \quad (2)$$

It is well known that an iteration of the elementary Gauss-Seidel method can be viewed as a set of projection methods with $\mathcal{L} = \mathcal{K} = \{e_i\}$, $i = 1, 2, \dots, n$, where e_i is the i th column of the identity matrix. In fact, a single correction is made at each step of these projection steps cycled for $i = 1, \dots, n$.

In [7], Ujević proposed a modification of the Gauss-Seidel method which may be named as a “one-dimensional double successive projection method” and referred to as 1D-DSPM. In an iteration of 1D-DSPM, a set of double successive projection methods with two pairs of one-dimensional subspaces are used. In fact, in an iteration of 1D-DSPM two pairs of subspaces $(\mathcal{K}_1, \mathcal{L}_1)$ and $(\mathcal{K}_2, \mathcal{L}_2)$ of one dimension are chosen while it makes double correction at each step of the process cycled for $i = 1, 2, \dots, n$. In [4], Jing and Huang proposed the “two-dimensional double successive projection method” and referred to as 2D-DSPM. In an iteration of 2D-DSPM, a set of projection methods with a pairs of two-dimensional subspaces \mathcal{K} and \mathcal{L} is used and a double correction at each step of the projection steps is made.

In this paper, a generalization of 2D-DSPM say m D-SPM which is referred to as “ m -dimensional successive projection method” is proposed and its convergence properties are studied. For $m = 2$, the m D-SPM results in 2D-DSPM.

Throughout this paper we use some notations as follows. By $\langle \cdot, \cdot \rangle$ we denote the standard inner product in \mathbb{R}^n . In fact, for two vectors x and y in \mathbb{R}^n , $\langle x, y \rangle = y^T x$. For any SPD matrix $M \in \mathbb{R}^{n \times n}$, the M -inner product is defined as $\langle x, y \rangle_M = \langle Mx, y \rangle$ and its corresponding norm is $\|x\|_M = \langle x, x \rangle_M^{1/2}$.

This paper is organized as follows. In section 2, a brief description of 1D-DSPM and 2D-DSPM are given. In section 3, the m D-SPM is presented and its convergence properties are studied. In section 4 the new algorithm and its practical implementations are given. Section 5 is devoted to some numerical experiments to show the efficiency of the method and comparing with 1D-DSPM and 2D-DSPM. Some con-

cluding remarks are given in 6.

2. A brief description of 1D-DSPM and 2D-DSPM

We review 1D-DSPM and 2D-DSPM in the literature of the projection methods. As we mentioned in the previous section in each iteration of 1D-DSPM a set of double successive projection method is used. Let x_k be the current approximate solution. Then the double successive projection method is applied as following. The first step is to choose two pairs of the subspaces $\mathcal{K}_1 = \mathcal{L}_1 = \{v_1\}$, $\mathcal{K}_2 = \mathcal{L}_2 = \{v_2\}$ and the next approximate solution x_{k+1} is computed as follows

$$\text{Find } \tilde{x}_{k+1} \in x_k + \mathcal{K}_1, \quad \text{such that} \quad b - A\tilde{x}_{k+1} \perp \mathcal{L}_1, \quad (3)$$

$$\text{Find } x_{k+1} \in \tilde{x}_{k+1} + \mathcal{K}_2, \quad \text{such that} \quad b - Ax_{k+1} \perp \mathcal{L}_2. \quad (4)$$

This framework results in [4, 7]

$$x_{k+1} = x_k + \alpha_1 v_1 + \beta_2 v_2$$

where

$$\alpha_1 = -p_1/a, \quad \beta_2 = (cp_1 - ap_2)/ad, \quad (5)$$

in which

$$a = \langle v_1, v_1 \rangle_A, \quad c = \langle v_1, v_2 \rangle_A, \quad d = \langle v_2, v_2 \rangle_A. \quad (6)$$

In [7], it has been proven that this method is convergent to the exact solution x_* of (1) for any initial guess.

In the 2D-DSPM, two two-dimensional subspaces $\mathcal{K} = \mathcal{L} = \text{span}\{v_1, v_2\}$ are chosen and a projection process onto \mathcal{K} and orthogonal to \mathcal{L} is used instead of double successive projection method used in 1D-DSPM. In other words, two subspaces $\mathcal{K} = \mathcal{L} = \text{span}\{v_1, v_2\}$ are chosen and a projection method is defined as following.

$$\text{Find } x_{k+1} \in x_k + \mathcal{K}, \quad \text{such that} \quad b - Ax_{k+1} \perp \mathcal{L}. \quad (7)$$

In [4], it has been shown that this projection process gives

$$\alpha = \frac{cp_2 - dp_1}{ad - c^2}, \quad \beta = \frac{cp_1 - ap_2}{ad - c^2},$$

where p_1 , p_2 , a , b , and c were defined in Eqs. (5) and (6). It has been proven in [4] that the 2D-DSPM is also convergent. Theoretical analysis and numerical experiments presented in [4] show that 2D-DSPM is more effective than the 1D-DSPM.

A main problem with 1D-DSPM and 2D-DSPM is to choose the optimal vectors v_1 and v_2 . In this paper, we first propose a generalization of 2D-DSPM and then give a strategy to choose vectors v_1 and v_2 in a special case.

3. m -dimensional successive projection method

Let $\{v_1, v_2, \dots, v_m\}$ be a set of m independent vectors in \mathbb{R}^n . For later use, let also $V_m = [v_1, v_2, \dots, v_m]$. Now we define the m D-SPM as follows. In an iteration of the m D-SPM we use a set of projection process onto $\mathcal{K} = \text{span}\{v_1, v_2, \dots, v_m\}$ and orthogonal to $\mathcal{L} = \mathcal{K}$. In other words, two m -dimensional subspaces \mathcal{K} and \mathcal{L} are used in the projection step instead of two two-dimensional subspaces used in 2D-DSPM. In this case Eq. (2) turns the form

$$\text{Find } y_m \in \mathbb{R}^m \text{ such that } x_{k+1} = x_k + V_m y_m, \quad \text{and} \quad V_m^T(b - Ax_{k+1}) = 0. \quad (8)$$

We have

$$\begin{aligned} r_{k+1} &= b - Ax_{k+1} \\ &= b - A(x_k + V_m y_m) \\ &= r_k - AV_m y_m, \end{aligned}$$

where $r_k = b - Ax_k$. Hence from Eq. (8) we deduce

$$0 = V_m^T r_{k+1} = V_m^T(r_k - AV_m y_m) = V_m^T r_k - V_m^T AV_m y_m \Rightarrow V_m^T AV_m y_m = V_m^T r_k.$$

The matrix $V_m^T AV_m$ is an SPD matrix, since A is SPD. Therefore

$$y_m = (V_m^T AV_m)^{-1} V_m^T r_k. \quad (9)$$

Hence, from (8) we conclude that

$$x_{k+1} = x_k + V_m (V_m^T AV_m)^{-1} V_m^T r_k. \quad (10)$$

Theorem 1. *Let A be an SPD matrix and assume that x_k is an approximate solution of (1). Then*

$$\|d_k\|_A \geq \|d_{k+1}\|_A, \quad (11)$$

where $d_k = x_* - x_k$ and $d_{k+1} = x_* - x_{k+1}$ in which x_{k+1} is the approximate solution computed by Eq. (10).

Proof. It can be easily verified that $d_{k+1} = d_k - V_m y_m$ and $Ad_k = r_k$ where y_m is defined by (7). Then

$$\begin{aligned} \langle Ad_{k+1}, d_{k+1} \rangle &= \langle Ad_k - AV_m y_m, d_k - V_m y_m \rangle \\ &= \langle Ad_k, d_k \rangle - 2\langle V_m y_m, r_k \rangle + \langle AV_m y_m, V_m y_m \rangle && \text{by } Ad_k = r_k \\ &= \langle Ad_k, d_k \rangle - \langle y_m, V_m^T r_k \rangle && \text{by (9)}. \end{aligned}$$

Therefore

$$\begin{aligned} \|d_k\|_A^2 - \|d_{k+1}\|_A^2 &= \langle Ad_k, d_k \rangle - \langle Ad_{k+1}, d_{k+1} \rangle \\ &= \langle y_m, V_m^T r_k \rangle \\ &= (V_m^T r_k)^T (V_m^T A V_m)^{-1} V_m^T r_k, && \text{by (9)} \\ &=: S(r_k). \end{aligned}$$

Since $(V_m^T A V_m)^{-1}$ is SPD then we have

$$\|d_k\|_A^2 - \|d_{k+1}\|_A^2 \geq 0,$$

and the desired result is obtained. \square

This theorem shows that if $V_m^T r_k = 0$ then $S(r_k) = 0$ and we don't have any reduction in the square of the A -norm of error. But, if $V_m^T r_k \neq 0$ then the square of the A -norm of error is reduced by $S(r_k) > 0$.

Theorem 2. *Assume that A is an SPD matrix and $\mathcal{L} = \mathcal{K}$. Then a vector x_{k+1} is the result of projection method onto \mathcal{K} orthogonal to \mathcal{L} with the starting vector x_k iff it minimizes the A -norm of the error over $x_k + \mathcal{K}$.*

Proof. See [6], page 126. \square

This theorem shows that if $\mathcal{K}_1 = \mathcal{L}_1 \subset \mathcal{K}_2 = \mathcal{L}_2$, then the reduction of A -norm of the errors obtained by the subspaces $\mathcal{K}_2 = \mathcal{L}_2$ is more than or equal to that of the

subspaces $\mathcal{K}_1 = \mathcal{L}_1$. Hence by increasing the value of m , the convergence rate may increase.

In the continue we consider the special case that the vectors v_i are the column vectors of the identity matrix. The next theorem not only proves the convergence of the method in this special case but also gives an idea to choose the optimal vectors v_i .

Theorem 3. *Let $\{i_1, i_2, \dots, i_m\}$ be the set of indices of m components of largest absolute values in r_k such that $i_1 < i_2 < \dots < i_m$. If $v_j = e_{i_j}$, $j = 1, \dots, m$ then*

$$\|d\|_A^2 - \|d_{new}\|_A^2 \geq \frac{m}{n\lambda_{\max}(A)} \|r_k\|_2^2. \quad (12)$$

Proof. Let $E_m = [e_{i_1}, e_{i_2}, \dots, e_{i_m}]$. By Theorem 1, we have

$$\begin{aligned} \|d_k\|_A^2 - \|d_{k+1}\|_A^2 &= S(r_k) \\ &= (E_m^T r_k)^T (E_m^T A E_m)^{-1} E_m^T r_k. \end{aligned}$$

Then by using Theorem 1.19 in [6] we conclude

$$S(r_k) \geq \lambda_{\min}((E_m^T A E_m)^{-1}) \|E_m^T r_k\|_2^2 \geq \frac{1}{\lambda_{\max}(E_m^T A E_m)} \|E_m^T r_k\|_2^2, \quad (13)$$

where for a square matrix Z , $\lambda_{\min}(Z)$ and $\lambda_{\max}(Z)$ stand for the smallest and largest eigenvalues of Z . It can be easily verified that [3]

$$\lambda_{\max}(E_m^T A E_m) \leq \lambda_{\max}(A), \quad \frac{(E_m^T r_k, E_m^T r_k)}{(r_k, r_k)} \geq \frac{m}{n}. \quad (14)$$

Hence

$$S(r_k) \geq \frac{m}{n\lambda_{\max}(A)} \|r_k\|_2^2, \quad (15)$$

and the desired result is obtained. \square

Eq. (12) shows the convergence of the method. Eq. (13) together with the first relation of the equation (14) give

$$S(r_k) \geq \frac{1}{\lambda_{\max}(A)} \|E_m^T r_k\|_2^2.$$

This equation gives an idea to choose indices i_j , $j = 1, \dots, m$. In fact, it shows that if these indices are the m components of the largest absolute values in r_k , then the lower bound of $S(r_k)$ depends on $\|E_m^T r_k\|_2^2$, and will be as large as possible.

In [3], another theorem for the convergence of the method obtained by $v_i = e_{i_j}$ was presented and an algorithm based upon this theorem was constructed for computing a sparse approximate inverse factor of an SPD matrix and was used as a preconditioner for SPD linear system.

4. Algorithm and its practical implementations

Hence, according to the results obtained in the previous section we can summarize the m D-DSM in the special case that $v_j = e_{i_j}$ as following.

Algorithm 1: m D-DSM

1. Choose an initial guess $x_0 \in \mathbb{R}^n$ to (1) and $r = b - Ax_0$.
2. Until convergence, Do
3. $x := x_0$
4. For $i = 1, \dots, n$, Do
5. Select the indices i_1, i_2, \dots, i_m of r as defined in Theorem 3
6. $E_m := [e_{i_1}, e_{i_2}, \dots, e_{i_m}]$
7. Solve $(E_m^T A E_m) y_m = E_m^T r$ for y_m
8. $x := x + E_m y_m$
9. $r := r - A E_m y_m$
10. EndDo
11. $x_0 := x$ and if x_0 has converged then Stop
12. EndDo

In practice, we see that the matrix $E_m^T A E_m$ is a principal submatrix of A with column and row indices in $\{i_1, i_2, \dots, i_m\}$. Hence, we do not need any computation for computing the matrix $E_m^T A E_m$ in step 7. For solving the linear system in step 7 of this algorithm one can use the Cholesky factorization of the coefficient matrix. Step 8 of the algorithm may be written as

- For $j = 1, 2, \dots, m$
- $x_{i_j} := x_{i_j} + (y_m)_j$
- EndDo

Hence only m components of the vector x are modified. Step 9 of the algorithm can be written as

$$r = r - \sum_{j=1}^m (y_m)_j a_{i_j}$$

where a_{i_j} is the i_j column of the matrix A .

It can be seen that Algorithm 2 in [4] is a special case of this algorithm. In fact, if $m = 2$ and the indices i_1 and i_2 are chosen as $i_1 = i$ and $i_2 = i - ij_{gap}$ ($i_2 = i - ij_{gap} + n$ if $i \leq ij_{gap}$), where ij_{gap} is a positive integer parameter less than n , then Algorithm 2 in [4] is obtained.

As we see, the first advantage of our algorithm over Algorithm 2 in [4] is that our algorithm chooses the indices $\{i_1, i_2, \dots, i_m\}$, automatically. Another advantage is that our algorithm chooses the indices such that the reduction in the square of the A -norm of the error is more than that of Algorithm 2 in [4]. Numerical experiments in the next section also confirm also this fact. The main advantage of Algorithm 2 in [4] over our algorithm is that only m components of the current residual are computed whereas in our algorithm the residual vector should be computed for choosing m indices of largest components in absolute value.

5. Numerical experiments

In this section we give some numerical experiments to compare our method with Algorithm 2 in [4]. Numerical results have been obtained by some MATLAB codes. We use all of the assumptions such as initial guess, exact solution, stopping criterion, and the examples used in [4]. Let $b = Ae$, where e is an n -vector whose elements are all equal to unity, i.e., $e = (1, 1, \dots, 1)^T$. We use $\|x_{k+1} - x_k\|_\infty < 10^{-6}$ as the stopping criterion. An initial guess equal to $x_0 = (x^1, \dots, x^n)$, where $x^i = 0.001 \times i$, $i = 1, 2, \dots, n$ is chosen. For each of the systems we give the numerical experiments of Algorithm 2 in [4] with $ij_{gap} = 2$ and 500 and our algorithm with $m = 2, 3, 4$ and 5.

Table 1: Results for the Example 1

Algorithm 2 in [4]	2D-DSM	3D-DSM	4D-DSM	5D-DSM
6 ($ij_{gap} = 2$)	5	4	3	2
7 ($ij_{gap} = 500$)				

Table 2: Results for the Example 2

Algorithm 2 in [4]	2D-DSM	3D-DSM	4D-DSM	5D-DSM
8 ($ij_{gap} = 2$)	7	6	4	4
9 ($ij_{gap} = 500$)				

Example 1. Let $A = (a_{ij})$ where

$$a_{ii} = 4n, \quad a_{i,i+1} = a_{i+1,i} = n, \quad a_{ij} = 0.5 \quad \text{for } i = 1, 2, \dots, n, \quad j \neq i, i+1.$$

We also assume $n = 1000$. Numerical experiments in terms of iteration number were shown in Table 1.

Numerical experiments presented in Table 1 show that the 2D-DSM method gives better results than the Algorithm 2 in [4]. This table also shows the effect of increase in m on the number of iterations for convergence.

Example 2. Let $A = (a_{ij})$ be the same matrix used in the previous example except the diagonal entries are changed to

$$a_{ii} = 3n, \quad i = 1, 2, \dots, n.$$

Numerical experiments were given in Table 2.

This table also shows the advantages of our method on Algorithm 2 [4].

Example 3. Our third set of test matrices used arise from standard five point finite difference scheme to discretize

$$-\Delta u + a(x, y)u_x + b(x, y)u_y + c(x, y)u = f(x, y), \quad \text{in } \Omega = [0, 1] \times [0, 1],$$

where $a(x, y)$, $b(x, y)$, $c(x, y)$ and $d(x, y)$ are given real valued functions. We consider three following cases:

Table 3: Results for the Example 3

Cases	Algorithm 2 in [4]	2D-DSM	3D-DSM	4D-DSM	5D-DSM
Case 1	391 ($ij_{gap} = 2$)	226	153	116	94
	323 ($ij_{gap} = 500$)				
Case 2	312 ($ij_{gap} = 2$)	192	131	100	80
	256 ($ij_{gap} = 500$)				
Case 3	302 ($ij_{gap} = 2$)	218	151	115	93
	250 ($ij_{gap} = 500$)				

Case1 : $a(x, y) = 0$, $b(x, y) = 10(x + y)$, $c(x, y) = 10(x - y)$, $f(x, y) = 0$,

Case2 : $a(x, y) = -10(x + y)$, $b(x, y) = -10(x - y)$, $c(x, y) = 1$, $f(x, y) = 0$,

Case3 : $a(x, y) = 10e^{xy}$, $b(x, y) = 10e^{-xy}$, $c(x, y) = 0$, $f(x, y) = 0$.

We assume $m = 32$. In this case we obtain three SPD matrices of order $n = 32 \times 32$ [4] and used them as the coefficient of the linear systems. Numerical results were given in Table 3.

This table also confirm that our method is more effective than the Algorithm 2 [4].

5. Conclusion

In this paper a generalization of the 2D-DSPM [4] which itself is a generalization of 1D-DSPM [7] is presented. 1D-DSPM and 2D-DPM need to prescribe some subspaces of \mathbb{R}^n for the projection steps. But our method in the spacial case chooses this subspaces automatically. Theoretical analysis and numerical experiments presented in this paper showed that our method is more effective than 2D-DSPM.

6. Acknowledgments

The author would like to thank Yan-Fei Jing for providing the matrices of Example 3.

References

- [1] R. Barrett et al., *Template for the solution of linear systems: building blocks for iterative methods*, SIAM Press: Philadelphia, 1994.
- [2] M. Benzi, *Preconditioning techniques for large linear systems: A survey*, J. Comput. Phys., 182 (2002) 418-477.
- [3] D. Khojasteh Salkuyeh and F. Toutounian, *A sparse-sparse iteration for computing a sparse incomplete factorization of an SPD matrix*, submitted.
- [4] Y.-F. Jing and T.-Z. Huang, *On a new iterative method for solving linear systems and comparison results*, J. Comput. Appl. Math., doi:10.1016/j.cam.2007.07.035, 2007.
- [5] C.D. Meyer, *Matrix analysis and applied linear algebra*, SIAM, 2004.
- [6] Y. Saad, *Iterative Methods for Sparse linear Systems*, PWS press, New York, 1995.
- [7] N. Ujević, *A new iterative method for solving linear systems*, Appl. Math. Comput. 179 (2006) 725730.